

This is a postprint version of the following published document:

Navarro Mesa, J. L., et al. (eds.) (2014). *Advances in Speech and Language Technologies for Iberian Languages: Second International Conference, IberSPEECH 2014, Las Palmas de Gran Canaria, Spain, November 19-21, 2014. Proceedings*. (pp. 109-118). (Lecture Notes in Computer Science; 8854). Springer International Publishing.
DOI: http://dx.doi.org/10.1007/978-3-319-13623-3_12

© 2014 Springer International Publishing Switzerland

Deep Maxout Networks Applied to Noise-Robust Speech Recognition

F. de-la-Calle-Silos, A. Gallardo-Antolín,
and C. Peláez-Moreno

Department of Signal Theory and Communications,
Universidad Carlos III de Madrid,
Leganés (Madrid), Spain
`fsilos@tsc.uc3m.es`

Abstract. Deep Neural Networks (DNN) have become very popular for acoustic modeling due to the improvements found over traditional Gaussian Mixture Models (GMM). However, not many works have addressed the robustness of these systems under noisy conditions. Recently, the machine learning community has proposed new methods to improve the accuracy of DNNs by using techniques such as dropout and maxout. In this paper, we investigate Deep Maxout Networks (DMN) for acoustic modeling in a noisy automatic speech recognition environment. Experiments show that DMNs improve substantially the recognition accuracy over DNNs and other traditional techniques in both clean and noisy conditions on the TIMIT dataset.

Keywords: noise robustness, deep neural networks, dropout, deep max-out networks, speech recognition, deep learning.

1 Introduction

Machine performance in Automatic Speech Recognition (ASR) tasks is still far away from that of humans, and noisy conditions only compound the problem. Noise robustness techniques can be divided into two approaches: feature enhancement and model adaptation. *Feature enhancement* tries to remove noise from the speech signal without changing the acoustic model parameters while *model adaptation* changes these parameters to fit the model to the noisy speech signal. Apart from these techniques, the last years have witnessed an important leap in performance with the introduction of new acoustic models based on Deep Neural Networks (DNNs) in comparison with conventional Gaussian Mixture Model-Hidden Markov Model (GMM-HMM) ([7], [3]) ASR systems. Nevertheless, the performance of these kind of ASR systems in noisy conditions has not yet been fully assessed.

Deep Neural Networks can be applied both in the so-called *tandem* [16] and *hybrid* [15] architectures. In the first case, DNNs can be trained to generate bottleneck features which are fed to a conventional GMM-HMM back-end. In the second, DNNs are employed for acoustic modeling by replacing the GMMs into an HMM system. In this paper we adopt a DNNs hybrid configuration.

DNN-HMM hybrid systems combine several features that make them superior to previous Artificial Neural Network (ANN)-HMM hybrid systems [11]: a) DNNs have a larger number of hidden layers leading to systems with many more parameters than the later. As a result, these models are less influenced by the mismatch between training and testing data but can easily suffer from overfitting if the training set is not big enough, b) the network usually models senones (tied states) directly (although there might be thousands of senones), and c) long context windows are used. Although conventional ANN also take into account longer context window than HMM or are able to model senones, the key to the success of the DNN-HMM is the combination of these components. DNN-HMM systems with these properties are often named Context-Dependent Deep Neural Network HMM (CD-DNN-HMM).

However, the most remarkable difference with traditional neural networks is that a *pre-training* stage is needed to reduce the chance that the error back-propagation algorithm employed for training falls into a poor local minimum. Besides, some recent methods have been proposed to avoid overfitting and improve the accuracy of the networks, as for example, dropout [8] which randomly omits hidden units in the training stage. Another related technique is the so-called Deep Maxout Networks (DMNs) [5] that splits the hidden units at each layer into non-overlapping groups, each of them generating an activation using a max pooling operation. This way, DMNs reduces the size of the parameter space significantly making it very suited for ASR tasks where the training sets and input and output dimensions are normally quite large. For this reason, DMNs have been employed in low-resources speech recognition devices [14] boosting the performance over other methods. We hypothesize that DMNs can improve the recognition rates in noisy conditions given that they are capable to model the speech variability from limited data more effectively [14].

As mentioned before, the number of research works that test DNNs in noisy conditions is still small. Notably, [18] applies DNNs with dropout on the Aurora 4 dataset with encouraging results. Up to our knowledge, the present paper is the first to apply Deep Maxout Networks in combination with dropout strategies in a noisy speech recognition task demonstrating a substantial improvement of the recognition accuracy over traditional DNN and other traditional techniques.

The remainder of this paper is organized as follows: Section 2 introduces deep neural networks and their application under a hybrid automatic speech recognition architecture, Section 3 and Section 4 describe the dropout and maxout methods, respectively. Finally, our results are presented in Section 5 followed by some conclusions and further lines of research in Section 6.

2 Deep Neural Networks and Hybrid Speech Recognition Systems

A Deep Neural Network (DNN) is a Multi-Layer Perceptron (MLP) with a larger number of hidden layers between its inputs and outputs, whose weights are fully connected and are often initialized using an unsupervised pre-training scheme.

As a traditional MLP the feed-forward architecture can be computed as follows:

$$\mathbf{h}^{(l+1)} = \sigma(\mathbf{W}^{(l)}\mathbf{h}^{(l)} + \mathbf{b}^{(l)}), \quad 1 \leq l \leq L \quad (1)$$

where $\mathbf{h}^{(l+1)}$ is the vector of inputs to the $l + 1$ layer, $\sigma(x) = (1 + e^{-x})^{-1}$ is the sigmoid activation function, L is the total number of hidden layers, $\mathbf{h}^{(l)}$ is the output vector of the hidden layer l and $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are the weight matrix and bias vector of layer l , respectively.

Training a DNN using the well-known error back-propagation (BP) algorithm with a random initialization of its weight matrices may not provide a good performance as it may become stuck in a local minimum. To overcome this problem, DNN parameters are often initialized using an unsupervised technique as Restricted Boltzmann Machines (RBMs) [6] or Stacked Denoising Autoencoders (SDAs) [19]. Nevertheless, as it will be explained later in this paper, pre-training may not be necessary if some recently proposed anti-overfitting techniques are used.

2.1 Hybrid Speech Recognition Systems

In a hybrid DNN/HMM system, just as in classical ANN/HMM hybrids [1], a DNN is trained to classify the input acoustic features into classes corresponding to the states of HMMs, in such a way that, the state emission likelihoods usually computed with GMM are replaced by the likelihoods generated by the DNN.

The DNN estimates the posterior probability $p(s|\mathbf{o}_t)$ of each state s given the observation \mathbf{o}_t at time t , through a softmax final layer:

$$p(s|\mathbf{o}_t) = \frac{\exp(\mathbf{W}^{(L)}\mathbf{h}^{(L)} + \mathbf{b}^{(L)})}{\sum_{\bar{s}} \exp(\mathbf{W}^{(L)}\mathbf{h}^{(L)} + \mathbf{b}^{(L)})}. \quad (2)$$

In a hybrid ASR system, the HMM topology is set from a previously trained GMM-HMM, and the DNN training data come from the forced-alignment between the state-level transcripts and the corresponding speech signals obtained by using this initial GMM-HMM system.

In the recognition stage, the DNN estimates the emission probability of each HMM state. To obtain state emission likelihoods $p(\mathbf{o}_t|s)$, the Bayes rule is used as follows:

$$p(\mathbf{o}_t|s) = \frac{p(s|\mathbf{o}_t) \cdot p(\mathbf{o}_t)}{p(s)} \quad (3)$$

where $p(s|\mathbf{o}_t)$ is the posterior probability estimated by the DNN, $p(\mathbf{o}_t)$ is a scaling factor constant for each observation and can be ignored, and $p(s)$ is the class prior which can be estimated by counting the occurrences of each state on the training data.

3 Dropout

The most important problem to overcome in DNN training is overfitting. Normally this problem arises when we try to train a large DNN with a small training

set. A training method called *dropout* proposed in [8] tries to reduce overfitting and improves the generalization capability of the network by randomly omitting a certain percentage of the hidden units on each training iteration.

When dropout is employed, the activation function of Eq. (1) can be rewritten as:

$$\mathbf{h}^{(l+1)} = m^{(l)} \star \sigma \left(\mathbf{W}^{(l)} \mathbf{h}^{(l)} + \mathbf{b}^{(l)} \right), \quad 1 \leq l \leq L \quad (4)$$

where \star denotes the element-wise product, $m^{(l)}$ is a binary vector of the same dimension of $\mathbf{h}^{(l)}$ whose elements are sampled from a Bernoulli distribution with probability p . This probability is the so called *Hidden Drop Factor (HDF)* and must be determined over a validation set as it will be seen in Section 5.

As the *sigmoid* function has the property that $\sigma(0) = 0$, Eq. (4) can be rewritten as:

$$\mathbf{h}^{(l+1)} = \sigma \left(m^{(l)} \star \left(\mathbf{W}^{(l)} \mathbf{h}^{(l)} + \mathbf{b}^{(l)} \right) \right), \quad 1 \leq l \leq L \quad (5)$$

where dropout is applied on the inputs of the activation function, leading a more efficient way of perform dropout training.

Note that dropout is only applied in the training stage whereas on testing all the hidden units become active. Dropout DNN can be seen as an ensemble of DNNs, given that on each presentation of a training example, a different sub-model is trained and the sub-models predictions are averaged together. This technique is similar to bagging [2] where many different models are trained using different subsets of the training data, but in dropout each model is only trained in a single iteration and all the models share some parameters.

Dropout networks are trained with the standard stochastic gradient descent algorithm but using the forward architecture presented on Eq. (4) instead of Eq. (1). Following [13], we compensate the parameters in testing by scaling the weight matrices taking into account the dropout factor as follows:

$$\overline{\mathbf{W}}^{(l)} = (1 - HDF) \cdot \mathbf{W}^{(l)} \quad (6)$$

Dropout has already successfully tested on noise robust ASR in [18]. Its benefits come from the improved generalization abilities attained by reducing their capacity. Another interpretation of the behaviour of dropout is that in the training state it adds random noise to the training set resulting in a network that is very robust to variabilities in the inputs (in our particular case, due to the addition of noise).

4 Deep Maxout Networks

A Maxout Deep Neural Network (DMN) [5] is a modification of the feed-forward architecture (Eq. (1)) where the maxout activation function is employed. The maxout unit simply takes the maximum over a set of inputs. In a DMN each

hidden unit takes the maximum value over the g units of a group. The output of the hidden node i of the layer $l + 1$ can be computed as follows:

$$h_i^{(l+1)} = \max_{j \in \{1, \dots, g\}} z_{ij}^{(l+1)}, \quad 1 \leq l \leq L \quad (7)$$

where $z_{ij}^{(l+1)}$ are the lineal pre-activation values from the l layer:

$$\mathbf{z}^{(l+1)} = \mathbf{W}^{(l)} \mathbf{h}^{(l)} + \mathbf{b}^{(l)} \quad (8)$$

As can be observed the max-pooling operation is applied over the $\mathbf{z}^{(l+1)}$ vector. Note that DMNs fairly reduce the number of parameters over DNNs, as the weight matrix $\mathbf{W}^{(l)}$ of each layer in the DMN is $1/g$ of the size of its equivalent DNN weight matrix. This makes DMN more convenient for ASR tasks where the training sets and the input and output dimensions are normally very large. An illustration of a DMN with 2 hidden layers and a group size of $g = 3$ is shown in Figure 1.

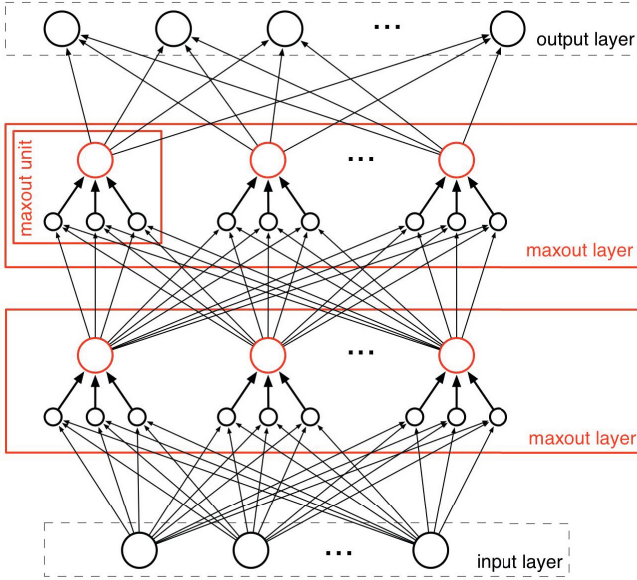


Fig. 1. A Maxout Network of 2 hidden layers and a group size of $g = 3$. The hidden nodes in red perform the max operation.

In [5] a demonstration of the capability of maxout units to approximate any convex function by tuning the weights of the previous layers is included. For this matter, the shapes of activation functions are not fixed allowing the DMNs to

model the variability of speech more smoothly. DMNs are commonly applied in conjunction with dropout maximizing the model averaging effects of dropout.

5 Experiments

In this section, we present the experiments carried out for evaluating and comparing the performance of conventional GMM-HMM and the different hybrid deep neural networks-based ASR systems (basic DNN, dropout DNN and DMN). The experiments were performed on the TIMIT corpus [4]; in particular, we used the 462 speaker training set, a development set of 50 speakers to tune all the parameters and finally the 24 speakers core test set. Each utterance is recorded at 16 kHz and the corpus includes time-aligned phonetic transcriptions allowing us to give results in terms of Phone Error Rate (PER).

To test the robustness of the different methods we digitally added to the clean speech four different types of noises (white, street, music and speaker) at four different SNRs using the FANT tool [9] (with G.712 filtering). These noises are the same ones used in [10]. All the noise tests are evaluated in a mismatch condition (i. e. training in clean conditions and testing on noisy speech).

On the technical side we employed the Kaldi toolkit [17] for implementing the traditional GMM-HMM ASR system and the PDNN toolkit [12] for the hybrid DNN-based ASR systems.

In all of the cases, the input features were 12th-order MFCCs plus a log-energy coefficient, and their corresponding first and second order derivatives yielding a 39 component feature vector. Mean and variance normalization on each of the components were applied. For the hybrid models, a context of 5 frames was chosen. All the hybrid systems were trained with the labels generated from the best performance GMM-HMM system through forced alignment.

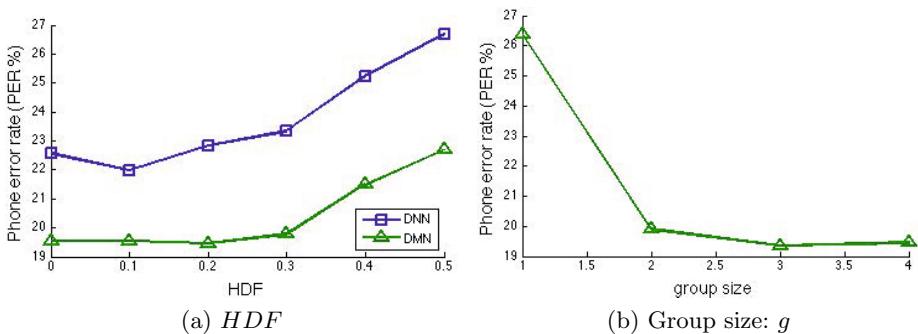


Fig. 2. Results in terms of PER [%] as a function of HDF for DNN and DMN (Figure 2a) and the group size for DMN (Figure 2b) on TIMIT development set. Both nets have 5 layers.

First, we tuned the configuration parameters of the networks (number of hidden layers, HDF and group size, when applicable) under clean conditions. HDF and group size were validated on the development set as can be seen on Figure 2 considering 5 hidden layer networks, yielding an optimal dropout factor of 0.1 for dropout DNNs, 0.2 for DMNs and a group size of $g = 3$. These values of HDR and group size were used throughout the rest of the experiments. DMNs are always employed in conjunction with dropout.

Figure 3 shows the PERs as a function of the number of hidden layers for the development and test sets for different types of hybrid DNN-based ASR systems: randomly initialized, with a pre-training stage, with dropout and maxout networks. The number of hidden nodes in all of the DNNs is 1024. To be fair, we chose 400 hidden maxout units for the DMN since $400 \times 3 = 1200$ yields a number of parameters in the same order as the DNNs. An exploration of the learning rates for the networks without dropout the learning rate started at 0.08 for 30 epochs and was subsequently divided in half while the validation error decreased. For the dropout and DMNs networks we started with a higher learning rate of 0.1. As can be seen in Figure 3 the DMNs outperform clearly the other networks for all the number of layers considered. Best results are obtained in the development set with DMNs of 5 layers.

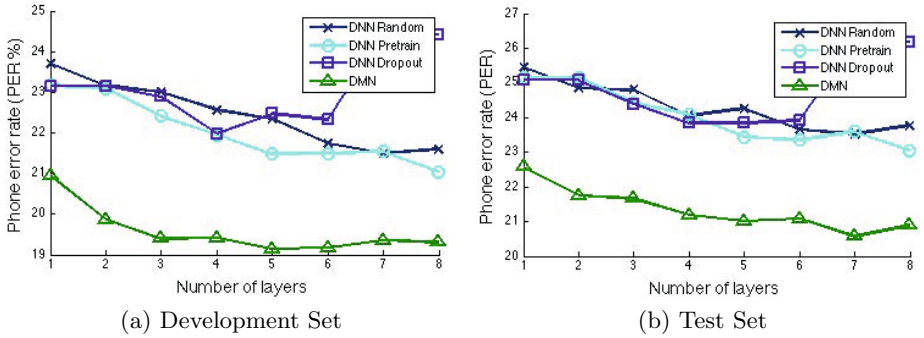


Fig. 3. Comparison of the performance of the different hybrid DNN-based ASR systems in terms of PER [%] as a function of the number of hidden layers for TIMIT development and test sets

Second, we compared the baseline system (GMM-HMM) with the best configuration of the different hybrid ASR systems under clean conditions: Monophone, Triphone, Triphone with Linear Discriminant Analysis (LDA), Maximum Likelihood Linear Transform (MLLT) and Speaker Adaptive Training (SAT). Results for the development and test sets are shown in Table 1.

As can be observed, all of the hybrid systems outperform the different versions of the baseline system, in both development and test sets. DNNs with random initialization, pretraining and dropout achieve similar results whereas with DMN the lowest PER is obtained.

Table 1. Recognition results in terms of PER(%) for the TIMIT development and core test sets in clean conditions

Method	Dev PER %	Eval PER %
Monophone	33.33	34.30
Triphone	28.64	30.42
Triphone + LDA + MLLT	26.44	27.62
Triphone + LDA + MLLT + SAT	23.56	25.79
DNN with random initialization (7 layers)	21.50	23.53
DNN with pretraining (8 layers)	21.05	23.05
DNN with dropout (4 layers)	21.98	23.84
DMN (5 layers)	19.15	21.01

Table 2. Average fine-tuning epoch execution time for a 5 hidden layers networks, 1024 nodes for DNN, 400 nodes and $g = 3$ for DMN

Method	Time(min)
DNN	57.81
DNN with dropout	59.07
DMN	24.10

Third, we tested the different systems in noisy conditions. Results achieved by the Monophone baseline, the best Triphone baseline (LDA+MLLT+SAT) and the best configurations for the hybrid DNN with pre-training, DNN with dropout, and DMN-based ASR systems in the noisy contaminated version of the TIMIT core test set are shown in Figure 4 for the different types of noises and four different SNRs. As can be seen, DMN performs better in almost every situation for white, street and speaker noises in comparison to the other systems. It is specially remarkable the performance of DMN in white and speaker noises. For street noise, results obtained with DMN are very similar to those achieved by the triphone GMM-HMM systems and both DNNs at high and medium SNRs whereas it obtains the lowest PER at low SNRs. For music noise, the results of all of the systems are very similar. As expected dropout performs better than DNN with pre-training at low SNR in all the noises, given that dropout is very robust to the variations of the input.

Fourth, we compared the fine-tuning stage time requirements for the DNNs, DNNs with dropout and DMN. We computed the average epoch time over all the iterations for 5 hidden layers networks with 1024 nodes per layer for the DNNs and 400 maxout units per layer and group size $g = 3$ for the DMN. The resulting times are shown in Table 2. As can be observed the DMN reduce the average epoch time over a half compared with DNNs with and without dropout. making them appealing for ASR tasks where the training set are normally very large.

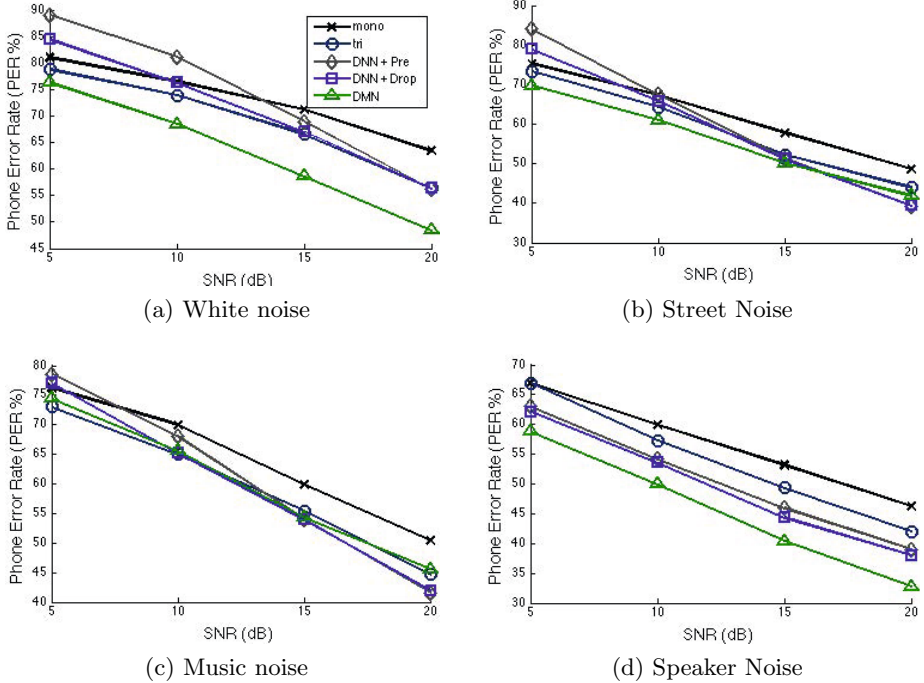


Fig. 4. Comparison of the performance of the different systems in terms of PER [%] for TIMIT test set in different noisy conditions

6 Conclusions and Future Work

In this paper Deep Maxout Networks (DMNs) are employed for robust speech recognition using hybrid architecture showing a better performance over standard DNNs. This is due to the DMNs flexibility of the activation functions allowing a better modeling of speech variability. Further lines of research include testing the DMN in a more complete datasets. Other novel machine learning techniques like dropconnect [20] are also interesting candidates not yet been tested in ASR tasks.

Acknowledgements. This contribution has been supported by an Airbus Defense and Space Grant (Open Innovation - SAVIER) and Spanish Government-CICYT project 2011-26807/TEC. We would also like to thank Chanwoo Kim for kindly providing the testing noises.

References

1. Bourlard, H., Morgan, N.: Connectionist Speech Recognition: A Hybrid Approach. Kluwer international series in engineering and computer science: VLSI, computer architecture, and digital signal processing. Springer US (1994)

2. Breiman, L.: Bagging predictors. *Machine Learning* 24(2) (1996)
3. Dahl, G.E., Yu, D., Deng, L., Acero, A.: Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech & Language Processing* 20(1) (2012)
4. Garofolo, J.S., Lamel, L.F., Fisher, W.M., Fiscus, J.G., Pallett, D.S., Dahlgren, N.L.: DARPA TIMIT acoustic phonetic continuous speech corpus cdrom (1993)
5. Goodfellow, I.J., Warde-Farley, D., Mirza, M., Courville, A., Bengio, Y.: Maxout Networks. *ArXiv e-prints* (2013)
6. Hinton, G.E.: A practical guide to training restricted boltzmann machines. In: Montavon, G., Orr, G.B., Müller, K.-R. (eds.) *NN: Tricks of the Trade*, 2nd edn. LNCS, vol. 7700, pp. 599–619. Springer, Heidelberg (2012)
7. Hinton, G.E., Deng, L., Yu, D., Dahl, G.E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., Kingsbury, B.: Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process. Mag.* 29(6) (2012)
8. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Improving neural networks by preventing co-adaptation of feature detectors. *CoRR* (2012)
9. Hirsch, G.: Fant - filtering and noise adding tool (2005), <http://dnt.kr.hsnr.de/download.html>
10. Kim, C., Stern, R.M.: Power-normalized cepstral coefficients (PNCC) for robust speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*
11. Li, J., Deng, L., Gong, Y., Haeb-Umbach, R.: An overview of noise-robust automatic speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22(4) (April 2014)
12. Miao, Y.: Kaldi+PDNN: Building DNN-based ASR systems with Kaldi and PDNN. *CoRR* (2014)
13. Miao, Y., Metze, F.: Improving low-resource CD-DNN-HMM using dropout and multilingual DNN training. In: *INTERSPEECH*, pp. 2237–2241. ISCA (2013)
14. Miao, Y., Metze, F., Rawat, S.: Deep maxout networks for low-resurce speech recognition. In: *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, Olomouc, Czech Republic, December 8-12 (2013)
15. Mohamed, A., Dahl, G.E., Hinton, G.E.: Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech & Language Processing* 20(1) (2012)
16. Morgan, N.: Deep and wide: Multiple layers in automatic speech recognition. *IEEE Transactions on Audio, Speech & Language Processing* 20(1) (2012)
17. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hanne-mann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K.: The Kaldi speech recognition toolkit. In: *IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society (2011)
18. Seltzer, M.L., Yu, D., Wang, Y.: An investigation of deep neural networks for noise robust speech recognition. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (2013)
19. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* 11, 3371–3408 (2010)
20. Wan, L., Zeiler, M.D., Zhang, S., LeCun, Y., Fergus, R.: Regularization of neural networks using dropconnect. In: *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, June 16-21 (2013)*